[文章编号] 1003-4684(2008)02-0044-03

# 基于 VC 的 STL 文件读取

张贞贞1,陈定方2

(1 武汉理工大学计算机科学学院, 湖北 武汉 430063; 2 武汉理工大学物流工程学院, 湖北 武汉 430063)

[摘 要] 就 CAD 系统较常用文件格式 ST L 的快速读取问题展开讨论, 分析了 ST L 两种文件格式的特点, 并讨论了冗余数据的消除.

[关键词] STL 文件; ASC II; 二进制

[中图分类号] TP391

[文献标识码]: A

STL(Stereo Lithography interface specification)文件格式<sup>[1]</sup>最初出现于 1989 年美国 3D SYS-TEM 公司生产的 SLA 快速成型系统,是一种应用于 CAD模型与成型系统之间数据转换的文件格式,现在已为大多数 CAD 系统和快速成型系统制造商所接受和采用,并且已经在快速成型技术领域中被誉为工业标准.本文旨在研究如何高效率读取 STL 文件.

## 1 STL 文件格式的结构

ST L 类似于实体数据模型的表面有限元网格划分. ST L 模型的数据通过给出组成三角形法向量的 3 个分量及三角形的 3 个顶点坐标来实现, ST L 文件记载了组成实体模型的所有三角形面片的法向量数据和顶点坐标数据, 有文本文件(ASCII)和二进制文件(BINARY)两种格式.

#### 1.1 STL的 ASCII 文件格式

ASCII 码格式 STL 文件逐行给出三角面片的几何信息,每一行以 1 个或 2 个关键字开头.在 STL 文件中的三角面片的信息单元 facet 是一个带矢量方向的三角面片, STL 模型就是由一组这样的三角面片构成.在一个 STL 文件中,每一个 facet 由7 行数据组成,第 1 行是三角面片指向实体外部的法向失量数据,第 2 行说明随后的第 3, 4, 5 行数据分别是三角面片的 3 个顶点信息,沿指向实体外部的法向矢量(第 1 行数据)方向逆时针排列.

ASCII<sup>[2]</sup>格式的 STL 文件结构如下:

endsolid filename stl

∥整个文件结束

#### 1.2 STL 的二进制文件格式

二进制 STL 文件用固定的字节数来给出三角面片的几何信息. 文件的起始 80 字节是文件头存储零件名,可以放入任何文字信息; 紧随着用 4 个字节的整数来描述实体的三角面片个数, 后面的内容就是逐个给出每个三角面片的几何信息. 每个三角面片占用固定的 50 字节, 它们依次是 3 个 4 字节浮点数, 用来描述三角面片的法矢量; 3 个 4 字节浮点数, 用来描述第 1 个顶点的坐标; 3 个 4 字节浮点数, 用来描述第 2 个顶点的坐标; 3 个 4 字节浮点数, 用来描述第 3 个顶点的坐标, 每个三角面片的最后 2 个字节用来描述三角面片的属性信息(包括颜色属性等)暂时没有用. 一个二进制 STL 文件的大小为三角形面片数乘以 50 再加上 84 个字节.

比较这 2 种方式可知, 二进制格式文件较小(通常是 ASC II 码格式的 1/5)<sup>[3]</sup>, 节省存储空间, 但 ASC II 码格式的文件可读性更强, 能被人工识别并被修改.

<sup>[</sup>收稿日期] 2007-12-01

## 2 STL 文件的读取与显示

STL文件是由一系列的三角形面片无序排列组合在一起的,没有反映三角形面片之间的拓扑关系,而且每一个小三角形面片必须与相邻的三角形面片共用2个顶点,每个顶点通常被重用6次左右,文件数据重复很多,造成了STL文件在读取、存储等方面的效率问题.本文研究了如何消除冗余数据并快速读取STL文件,这对于后续三维图形的现实.是一个重要和基本的环节.

去除冗余点的实质是,从STL文件中依次取出 三角形,通过比较三角形顶点的坐标值,归并重复的 点,从而建立一个无重复点的点表,去除冗余点最直 接的方法就是比较法. 对于新的顶点, 读进来时依次 对X 坐标、Y 坐标和Z 坐标进行比较,且保证所有顶 点依次按照X 坐标、Y 坐标和Z 坐标排序. 这样大大 减少了查找范围,提高了排序和插入的效率,采用上 述分析方法建立点表,有3个步骤:第一步,按三角 形为单元,从文件中读入该三角形的3个顶点,逐点 进行比较. 第二步, 判断该顶点是否已经在点表中存 在. 1) 进行 X 坐标的比较. a)X 坐标相同的顶点不 存在, 转入第三步. b)X 坐标相同的顶点存在, 转入 (2), (2) 在所有 (X) 坐标相同的顶点中, 比较 (Y) 坐标. a) Y 坐标相同的顶点不存在, 转入第三步. 3) Y 坐标 相同的顶点存在, 转入4). 4) 在所有X 坐标和Y 坐 标都相同的顶点中, 比较 Z 坐标. a)Z 坐标相同的顶 点不存在, 转入第三步. b) Z 坐标相同的顶点存在, 说明该顶点已经在存在于点表中所以新的顶点不加 入点表中. 第三不, 向点表中插入新的顶点, 同时完 成排序. 这样建立无冗余点的点表, 效率很高, 而且 在搜索比较的同时, 具有相同坐标值 $(X, Y \to Z)$  的 顶点自动排在一起,实现了自动分段功能,提高了后 续搜索效率,实现上面的步骤.

为实现上述步骤使用的 Visual C+ + 数据结构

```
如下:
struct Vertex{
double x;
                                    ∥ 顶点的 X 坐标值
double v;
                                    ∥ 顶点的 Y 坐标值
double z;
                                    ∥ 顶点的 Z 坐标值
vertex * x_s am e;
                       ∥ 指向 X 坐标值相同的下一个顶点
vertex * y_same;
               ∥ 指向 X 坐标和 Y 坐标值都相同的下一个顶点
vertex* next;
                      ∥ 指向 X 坐标值不相同的下一个顶点
} vertex;
struct Facet{
double x;
                              ∥面片法向量的 X 坐标值
```

∥面片法向量的 Y 坐标值

∥ 面片法向量的 Z 坐标值

double y;

double z: 1994-2012 China Academic

```
vertex * vertex 1;#指向该面片的第一个顶点vertex * vertex 2;#指向该面片的第二个顶点vertex * vertex 3;#指向该面片的第三个顶点} facet;
```

使用上述方法对所读入的 STL 文件进行处理后,去除冗余点的同时也建立了点表,表中坐标值 X 升序排列,而每个 X 值后有一个或多个 Y 值,其中 Y 值也按升序排列;同理,每个 Y 值后有一个或多个 Z 值,其中 Z 值也是按升序排列的.图 1 给出了点表形成后的逻辑结构图.

图 1 点表的逻辑结构图

如图 1 所示,所有顶点以多重链表<sup>[4]</sup> 形式连接: 顶点  $v_1, v_2, v_3, v_4$  形成的链中各自的 X 坐标值互不相同;  $v_1, v_5, v_8$  形成的链中各自的 X 坐标相同, Y 坐标互不相同;  $v_5, v_7$  形成的链中各自的 X 坐标和 Y 坐标都相同, Z 坐标不同;  $v_2, v_6$  形成的链类似于  $v_5, v_7$  形成的链, 其中各自的 X 坐标和 Y 坐标都相同, Z 坐标不同.

此外, 由结构体 facet 形成的链表(面表) 记录了每个面片的法向量以及它的 3 个顶点的信息, 使得面表和点表联系在一起, 后续要显示三维图形时顺序读取面表, 通过面表间接在点表中找到各顶点.

这样,整个STL 文件所有三角面片的数据可由 facet 结构的一个数组 m\_facet 来记录. 为实现动态存储,该数组定义如下:

CArray< facet , facet> m\_facet;

用指针 vertex\* head 作为点表的头指针, 判断 一个顶点是否在点表中存在具体算法如下:

```
vertex* Sort(vertex* v, vertex* head) {
vertex* p; veretex* q;

# 链表为空时插入点 v
if(head == NULL) {head = v; return v; }
else{p = head; q = NULL;
while(p) {
if(v-> x == p-> x) {
# 在 X 坐标值相同的链中查找
while(p) {
if(v-> y == p-> y) {
# 在 X, Y 坐标值都相同的链中查找
while(p) {
# 点表中已存在点 v, 返回其位置
if(v-> z == p-> z) {return p; }
else{ # 在 X, Y 坐标值相同的链中查找并插入点 v
```

ıblisif(ng House-Alphyntspreserved> y-same;//www.cnki.net

```
els e{
if(q = NULL) \{ head = v; \}
if(q- > next = = p) \{q- > next = v; \}
if(q- > x_same = = p) \{q- > x_same = v; \}
if(q- > y_same = = p) \{q- > y_same = v; \} 
v- > y_same = p; v- > x_same = p- > x_same;
p- > x_{same} = NU LL; v- > next = p- > n ext;
p- > n ext = NU LL; return v; \} \}
if(p = NULL) \{q- > y\_sam e = v; return v; \}\}
els e{ | 在 X 坐标值相同的链中查找并插入点 v
if(v- > y > p- > y) \{q = p; p = p- > x\_same; \}
els e{
if(q = NULL) \{ head = v; \}
els e{
if(q- > next = = p) \{q- > next = v; \}
if(q- > x_same = = p) \{q- > x_same v; \} 
v- > x_sam e = p; v_next = p- > n ext;
p- > n ext = NU LL; return v; \} \}
if(p = NULL) \{q- > x\_same = v; return v; \}\}
if(v- > x > p- > x) \{ q = p; p = p- > next; \}
if(q = NULL) \{ head = v; \}
els e{q- > nex t = v;}
v- > n ext = p; return v; \} \}
if(p = = NU LL) \{q- > next = v; return v; \}\}
     然后读取 A SC II 形式的 ST L 文件, 存储文件
信息,具体算法如下:
```

```
BOOL LoadST LFile(LPCT ST R stlfile) {
FILE* file;
if((file = fopen(stlfile, "r")) = = NULL) return false;
char str[ 80];
vertex* v1, v2, v3;
facet* f;
while (fscanf(file, "\% s", str) = = 1)
if(strncmp(str, "normal", 6) = = 0) {
v1= (vertex*) malloc(sizeof(vertex));
v2= (vertex*) malloc(sizeof(vertex));
v3= (vertex*) malloc(sizeof(vertex));
f = (facet*) malloc(sizeof(facet));
| 读取面片的法向量
f\,s\,canf(\,f\,ile,\,\text{\tt ''}\,\%\,\,1f\,\,\%\,\,1f\,\,\%\,\,1f\,\text{\tt ''}\,,\,\&\,\,(\,f-\,>\,x)\,\,,\,\&\,\,(\,f-\,>\,y)\,\,,\,\&\,\,(\,f-\,>\,z)\,\,)\,\,;
fscanf(file, "% * s % * s");
∥ 读取面片的三个顶点信息到 v1, v2, v3
fs canf(file, "%* s % 1f % 1f % 1f", & (v1->x), & (v1->y), &
(v1- > z));
fs canf(file, "%* s % 1f % 1f % 1f", & (v2->x), & (v2->y), &
(v2->z)):
(v3- > z));
```

# 分别判断 v1, v2, v3 是否在点表中出现过并完成而表到点表的映射

```
f- > vetex1= Sort(v1, head);

f- > vetex1= Sort(v2, head);

f- > vetex1= Sort(v3, head);

m_facet. Add(f); }}
```

#### 最后,通过顺序读取面表 m\_facet 可以获取所

```
有三角面片信息,在 VC 中利用 OpenGL<sup>[5]</sup> 来实现三维图形的显示. 具体算法如下:
```

```
for (int i = 0; i < number; i+ +) {
  glBegin(GL. TRIANGLES);
  glNormal3f(m_triangle[i].x, m_triangle[i].y, m_
  triangle[i].z);
  glVertex3f(m_triangle[i].vertex1- > x, m_triangle[i].vertex1- > y, m_triangle[i].vertex1- > z);
  glVertex3f(m_triangle[i].vertex2- > x, m_triangle[i].vertex2- > y, m_triangle[i].vertex2- > z);
  glVertex3f(m_triangle[i].vertex3- > x, m_triangle[i].vertex3- > x, m_triangle[i].vertex3- > y, m_triangle[i].vertex3- > z);
  glVertex3f(m_triangle[i].vertex3- > x, m_triangle[i].vertex3- > z);
  glEnd();
}
```

函数 glBegin<sup>[5]</sup> 实用参数 GL. TRIANGLES 将产生实体造型效果,即真实感图形显示.

## 3 结语

STL 文件中每个顶点通常被重用 6 次左右, 因而文件数据重复很多. 通过消除冗余顶点信息可以极大地减少存储空间, 提高对 STL 文件的读取和存储效率, 为三维物体的建模提供了方便. 此外, 采用多重链表形式存储定点信息在消除冗余顶点的同时, 使坐标值相同的点连在一起, 并且按照坐标值的升序排列, 减少了在查找新的顶点是否在点表中存在时的比较次数, 提高了查找效率.

#### [参考文献]

- [1] 张红玲.STL 格式实体真实感图形显示工具[J]. 电脑 学习,2000(4):31-32.
- [2] 王清辉,王 彪. Visual C+ + CAD 应用程序开发技术[M]. 北京: 机械工业出版社, 2003.
- [3] 卫 炜, 周来水, 张丽艳. 海量 STL 文件的快速读取与显示[J]. 机械科学与技术, 2006, 25(6):935-938.
- [4] 严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版 社, 2007.
- [5] Shreiner D, Woo M, Neider J, et al. OpenGL 编程指南 [M]. 徐 波译. 北京: 人民邮电出版社, 2005.

(下转第73页)

#### 参考文献

- [1] 申忠如,郭福田,丁 晖. 现代测试技术与系统设计 [M]. 西安交通大学出版社,2006: 102-106.
- [2] 赵浪涛. BP 神经网络 PID 控制器研究[J]. 兰州工业高 等专科学校学报, 2007, 14(2): 34-35.
- [3] 凌振宝,王 君,朱凯光. 数字温度传感器在热电偶冷端温度补偿中的应用[J]. 传感器技术, 2003, 22(6): 45 46.

## The Research on the Intelligent Instrument for Measuring Temperature Based on PID Neural Network

ZHU Ke<sup>1</sup>, LIU Hong-li<sup>1</sup>, ZHEN Yu-yun<sup>1,2</sup>

(1 The School of Automation, Wuhan Univ. of Technology, Wuhan 430063, China; 2 Institute of Intelligent Manufacturing, Wuhan Univ. of Technology, Wuhan 430063, China)

**Abstract:** Thermocouple is widely applied in industry temperature measurement, but it is not a linear relation between its temperature and potential, which limits the precision of measuring temperature. In order to solve the problem of the accuracy of thermocouple temperature calculation, the hardware circuit of Intelligent Instrument is designed with feed forward correction function of the PID Neural Network. The result shows that this computing method has good reliability and practicability.

Keywords: thermocouple; PID neural network; intelligent instrument for measuring temperature

[责任编校: 张 众]

(上接第46页)

## STL File Loading Based On VC

ZHANG Zhen-zhen<sup>1</sup>, CHEN Ding-fang<sup>2</sup>

(1 School of Computer Science, Wuhan Univ. of Technology, Wuhan 430063, China; 2 School of Logistics Engine, Wuhan Univ. of Technology, Wuhan 430063, China)

**Abstract:** STL has been the industry standard in the field of rapid prototype technology. How to load and visualize the STL file rapidly with deleting redundancy vertex in STL files is discussed in this paper in the aspects of STL rapid processing of common file formats and the analysis of two STL file formats.

**Keywords:** STL; ASCII; binary

[责任编校:张 众]